

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

**Rozšíření algoritmu optimalizace
přeprav o data z modulu AETR a vstup
z rozhraní modulu routingu**

**Extension of transport optimization
algorithm by AETR module data and
input from routing module interface**

Zadání diplomové práce

Student:

Bc. Petr Bojko

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Rozšíření algoritmu optimalizace přeprav o data z modulu AETR a vstup
z rozhraní modulu routingu

Extension of Transport Optimization Algorithm by AETR Module
Data and Input from Routing Module Interface

Jazyk vypracování:

čeština

Zásady pro vypracování:

Výsledkem práce bude rozšíření modulu optimalizace, který bude zohledňovat stav řidičů pomocí AETRu a pro optimalizaci tras bude využívat routing. Modul optimalizace přiřadí objednávky jen těm řidičům, kterým AETR dovolí danou objednávku zvládnout. Trasy mezi body objednávek budou zpracovány routingem. Na mapě se zobrazí trasy k jednotlivým stanovištím objednávek pomocí routingu.

1. Seznamte se s případem obchodního cestujícího.
2. Nastudujte modul optimalizace ve stávajícím systému GX.
3. Seznamte se s vyhláškou AETRu pro Evropu.
4. Nastudujte rozhraní modulu AETR ve stávajícím systému GX.
5. Navrhněte způsob začlenění AETRu do optimalizace.
6. Naimplementujte a otestujte zpracování dat AETRu v optimalizačním modulu.
7. Seznamte se s modulem routing v systému GX.
8. Navrhněte způsob začlenění routingu do optimalizace.
9. Naimplementujte a otestujte routing v optimalizačním modulu.
10. Celou funkčnost předved'te na datech zákazníka od GX.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

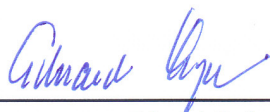
Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Marek Bober**

Konzultant diplomové práce: Ing. Jan Martinovič, Ph.D.

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. dubna 2016

.....

Souhlasím se zveřejněním této diplomové práce dle požadavků čl.26, odst.9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava

Dne 25.4.2017



.....
podpis



GX SOLUTIONS BOHEMIA, s. r. o.
V Oblouku 114, 251 01 Čestlice
IČ: 26850010 • DIČ: CZ26850010
Tel.: +420 225 396 333
www.gxsolutions.eu

Na tomto místě bych rád poděkoval především vedoucímu mé práce Ing. Marku Boberovi, za jeho odborné rady a trpělivost. Dále bych chtěl poděkovat své rodině, která mi byla obrovskou oporou.

Abstrakt

Práce je zaměřena na integraci zohlednění reálných vzdáleností mezi body objednávek do knihovny zpracující Vehicle Routing Problem a dále také na zajištění neporušení vyhlášky ES 561. Reálné vzdálenosti komunikací jsou získané pomocí routing služby IT4Innovations a vyhláška ES 561 je úpravou samotného předpisu AETR o maximální povolené době jízdy řidiče. Integrace tohoto algoritmu je provedena do monitorovacího systému GX-GO od společnosti GX Solutions Bohemia, s.r.o., který v rámci telematiky slouží dopravcům a popřípadě obyčejným uživatelům pro sledování jejich vozidel.

Klíčová slova: AETR, ES 561, routing, optimalizace, webová aplikace

Abstract

This thesis is focused on integration of real routing distances between favours points for Vehicle Routing Problem library and also on ensuring, that decree ES 561 will not be violated. Real routing distances are acquired using IT4Innovation routing services and decree ES 561 is modification of AETR regulation of driver's driving hours. This algorithm is integrated into monitoring system called GX-GO from GX Solutions Bohemia s.r.o company, which serves within telematics to carrier companies and regular user for monitoring of their vehicles.

Key Words: AETR, ES 561, routing, optimization, web application

Obsah

Seznam použitých zkratk a symbolů	10
Seznam obrázků	11
Seznam tabulek	12
Seznam výpisů zdrojového kódu	13
1 Úvod	14
2 Telematika	16
2.1 Výhody telematiky	16
3 GX-GO	18
4 Použité technologie	20
4.1 WCF	20
4.2 ASP.NET	20
4.3 ASP.NET MVC	20
4.4 IIS Express	20
4.5 Razor syntaxe	20
4.6 DevExpress pro ASP.NET MVC	21
4.7 PostgreSQL	21
4.8 OpenLayers	21
4.9 JavaScript a jQuery	22
4.10 Entity Framework	22
5 Příklad obchodního cestujícího	23
5.1 Metrický obchodní cestující	23
6 Vehicle routing problem	25
7 Současný stav optimalizačního modulu	27
8 Modul AETR v GX-GO	29
8.1 Tachograf	29
8.2 Vyhlášky AETR a ES 561	29
9 Routing v GX-GO	32
10 Popis postupu optimalizace a návrh začlenění celkového řešení	34

11 Návrh začlenění routingu do optimalizace	36
11.1 Depo	36
11.2 Vyloučení použití vzdálenostní matice	36
11.3 Model pro data routingu	36
11.4 Paralelní zisk dat routingu	36
11.5 Předání dat do VRPTUO	38
12 Návrh začlenění AETRu do optimalizace	40
12.1 Vyčítání řidičů	41
12.2 Datový model pro výsledek zohlednění AETRu	41
12.3 Dopočet budoucích dat o zbývajících jízdách	41
12.4 Nastavení časového okna řidiče	44
12.5 Úprava nastavení optimalizace	45
13 Testování na datech zákazníka GX Solutions	46
13.1 Rozdíly mezi optimalizací s a bez dat AETRu	46
13.2 Rozdíly mezi optimalizací s a bez dat routingu	47
13.3 Testování s reálnými objednávkami zákazníků	47
14 Omezení algoritmu	49
14.1 Velice rozlišná okna objednávek	49
14.2 Možné nepovolené navýšení jízdového času řidiče	50
14.3 Problém s různými rychlostmi vozidel	50
15 Možná rozšíření do budoucna	51
15.1 Zohlednění stylu jízdy řidiče	51
15.2 Překrývající se nové objednávky se starými	51
15.3 Zkrácení přestávky	51
16 Závěr	53
Literatura	54
Přílohy	55
A Příloha na CD/DVD	56

Seznam použitých zkratk a symbolů

ACID	– Atomicity, Consistency, Isolation, Durability
AETR	– Accord européen sùn les transports routiers - Evropská dohoda o práci osádek vozidel v mezinárodní silniční dopravě
AJAX	– Asynchronous JavaScript and XML
ASP	– Active Server Pages
ASPX	– Active Server Pages .NET
ES 561	– Nařízení Evropského parlamentu a rady č. 561/2006
GPS	– Global Positioning System
GSM	– Globam System for Mobile Communications
HTML	– HyperText Markup Language
IIS	– Internet Information Services
MB	– Megabyte
MVC	– Model View Controller
TSP	– Travelling Salesman Problem
VRP	– Vehicle Routing Problem
VRPTUO	– Vehicle Routing Problem knihovna od Vysoké školy báňské - Technická univezita Ostrava
WCF	– Windows Communication Foundation
Jsprit	– Knihovna řešící Vehicle Routing Problem založená na programovacím jazyce Java

Seznam obrázků

1	Mapový modul GX-GO	17
2	Mapový modul GX-GO	19
3	Příklad 2-aproximačního algoritmu	24
4	Modul optimalizace	28
5	Data získaná v rámci modulu AETR	31
6	Modul routingu	32
7	Návrh postupu řešení v základních bodech.	35
8	Třídní diagram bodu zájmu (depo/místo objednávky)	37
9	Návrh začlenění zohlednění ES 561	40
10	Varianta dopočítání zbývajících jízdního času pro současný týden	41
11	Varianta dopočítání zbývajících jízdního času pro současný týden	42
12	Varianta dopočítání zbývajících jízdního času pro příští týden	43
13	Varianta dopočítání zbývajících jízdního času pro příští týden	43
14	Posun časového okna k časově nejbližší následující objednávce	45
15	Nastavení modulu optimalizace	45
16	Optimalizované objednávky za použití AETRu	46
17	Optimalizované objednávky bez použití AETRu	46
18	Optimalizace trasy za pomoci dat z routingu	47
19	Trasa na mapě za pomoci dat z routingu	47
20	Optimalizace trasy za pomoci dat vzdušných vzdáleností	48
21	Trasa na mapě za pomoci dat vzdušných vzdáleností	48
22	Nedostatek v rámci nastavení časového okna	49
23	Zvolení vhodného časového okna	51
24	Zkrácení doby řidičova denního odpočinku	52

Seznam tabulek

1	Limity PostgreSQL	21
---	-----------------------------	----

Seznam výpisů zdrojového kódu

1	ASPX syntaxe	20
2	Razor syntaxe pro C#	21
3	Kód zisku dat routingu (C#)	32
4	Paralelní zisk dat routingu (C#)	37
5	Zisk patřičných vzdáleností mezi dvěma body (C#)	38
6	Zisk budoucího jízdního času (C#)	43

1 Úvod

Optimalizace trasy je v současné době důležitým aspektem, který ovlivňuje náklady dopravců. Z tohoto důvodu vznikl optimalizační algoritmus pojmenovaný Vehicle routing problem založený na Problému obchodního cestujícího.

V této práci jsou oba problémy stručně popsány, ale její hlavní zaměření je na rozšíření VRPTUO o zpracování dat o reálných vzdálenostech ujetých tras a jejich čase a také na určení optimálního časového okna řidičů na základě jejich odježděných hodin tak, aby nebyla porušena vyhláška ES 561.

Hlavní body této vyhlášky, která je úpravou vyhlášky AETR určenou pro země Evropské unie a body, které zohledňuje současný modul zaměřený na *ES 561* jsou v této práci popsány.

Dodržování této vyhlášky je pro bezpečnost na silnicích stěžejní. Jak vyplývá z průzkumu Federálního výzkumného institutu silnic (Bundesanstalt für Straßenwesen) odhadované procento vážných nehod kamiónů v Německu způsobených vyčerpáním řidičům je 19 procent. [17]

Zpracování dat z routingu do optimalizační třídy VRP od Vysoké školy Báňské a předem nutné vyřazení nevhodných řidičů bylo provedeno na základě požadavků společnosti GX Solutions Bohemia s.r.o. (dále jen GX Solutions) do jejich webové aplikace *GX-GO*.

GX Solutions je společnost založená v roce 1998. Hlavní centrála se nachází v Praze a provozna společnosti ve Frýdku-Místku. Jejím hlavním zaměřením je monitoring vozidel, především podnikových. Nejdříve poskytovala informace o využívání a stavu vozidel především u kamiónů. V současnosti mají stovky zákazníků a monitorují tisíce vozidel, včetně vlaků, autobusů, vysokozdvizných vozíků apod. Ze začátku byly informace získávány na základě off-line principu, poté, po vytvoření řešení se společností TOROLA electronic, spol. s.r.o., se společnost pustila do online monitoringu.

GX Solutions poskytuje svým zákazníkům služby jako GPS monitoring, telemetrická řešení, SW a HW řešení na míru zákazníka, odborné konzultace, poradenství, školení a servisní služby. Vize společnosti spočívá v tom, že chce v budoucnosti pokračovat v inovacích řešení. Také má za cíl zobrazit důležitost telemetrie v moderních IT řešeních, bez kterých, dle jejího názoru, nelze flotilu spravovat efektivně at' už jste v obchodě, službách, dopravě, stavebnictví nebo průmyslu.

V současné době patří mezi lídry monitoringu vozidel a strojů na českém trhu a v roce 2011 získala ocenění Nejlepší elektronický/telematický systém. Pro dispečery společností vyvinuli jak desktop, tak webovou aplikaci pro monitorování stavu vozidel. V jejich systému lze nalézt modul hlídání událostí, dispečerský modul, modul sestav a spoustu dalších funkcí.

Jako první bod pro zavedení daných funkcí byla nutná úprava stávajícího modulu optimalizace. Dále byl proveden přechod z Jsprit na VRPTUO, s tím související úprava zpracování výsledných dat. Poté bylo provedeno navržení zisku a zpracování dat získaných pomocí routingu IT4Innovations do VRPTUO s důrazem na co největší rychlost zisku dat a jejich zpracování. Stěžejním bodem této práce byla predikce stavu zbývajících času jízdy řidičů daných vozidel

v den optimalizovaných objednávek, a poté jejich předání do VRPTUO tak, aby bylo možné obsloužit co největší množství objednávek.

V závěrečných kapitolách jsou popsána omezení algoritmu, návrh jejich řešení 14 a možné kroky do budoucna, popřípadě návrh jejich začlenění do současného algoritmu. 15

2 Telematika

Výraz telematika vznikl spojením slov telekomunikace a informatika. Jedná se v podstatě o přenášení informací skrze telekomunikační síť a jejich zpracování za použití informačních technologií, týkajících se vzdálených objektů, v tomto případě vozidel.

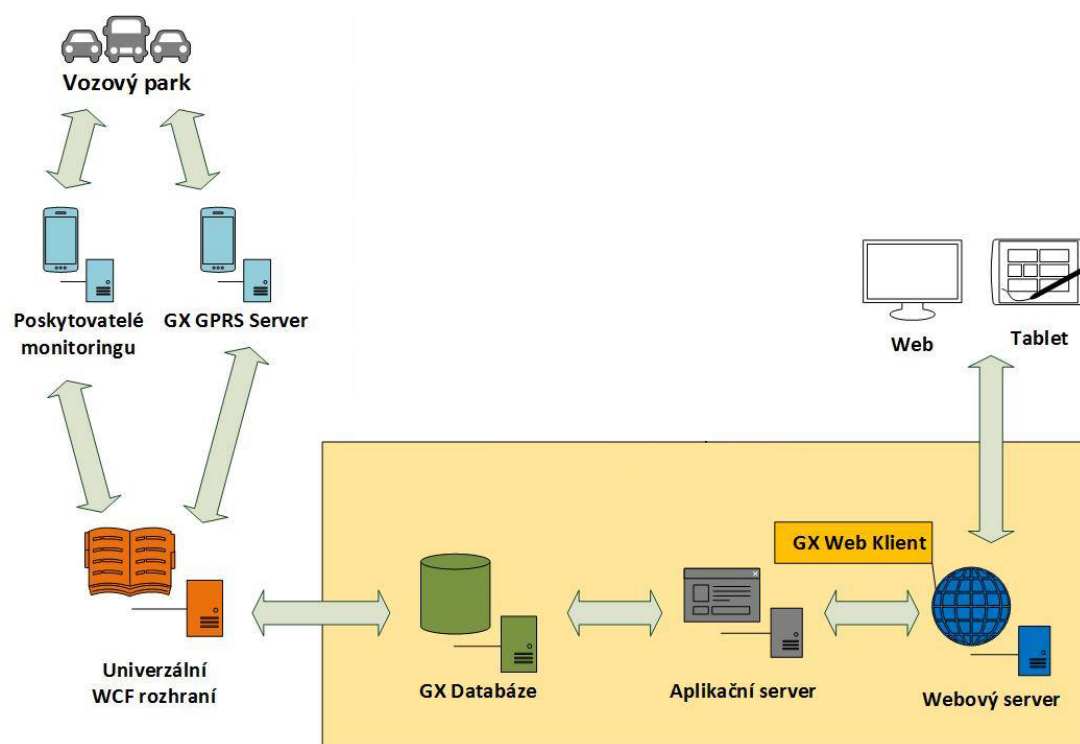
Slouží ke sledování pozice, pohybu, stavu a chování vozidla a to za pomoci GPS přijímače, GSM modulu a jednotky, která sleduje stav zařízení. Kromě těchto základních informací mohou být poskytovány i další informace, jako jsou přihlášený řidič, aktuální rychlost vozidla, prudkost brždění nebo rozjezdů a mnohé další. [1]

2.1 Výhody telematiky

Využití telematického řešení přináší dopravcům mnohé výhody, mimo jiné:

- Zvýšení produktivity,
- snížení ceny za práci,
- zlepšení zákaznických služeb,
- zvýšení bezpečnosti a zabezpečení
- a mnohé další.

Pro přibližnou ilustraci celé komunikace a zpracování dat, jak ve většině telematických řešení tento proces funguje viz 1. Obrázek konkrétně zobrazuje přibližný chod dat podle společnosti *GX Solutions*.



Obrázek 1: Mapový modul GX-GO

3 GX-GO

Jedná se o webovou aplikaci sloužící zákazníkům *GX Solutions*, kteří nepoužívají desktop aplikaci, ke sledování polohy a stavu jejich vozidel.

Vývoj aplikace začal v roce 2014 za účelem nahrazení původní webové aplikace, především z důvodu blížícího se ukončení podpory technologie *Silverlight*, kterou aplikace využívala. [3]

V současné době není aplikace používána všemi zákazníky, kteří nepoužívají desktop aplikaci, jelikož část z nich používá stále starší webovou aplikaci, ale v následujících týdnech se očekává přechod celé zákaznické webové skupiny na *GX-GO*.

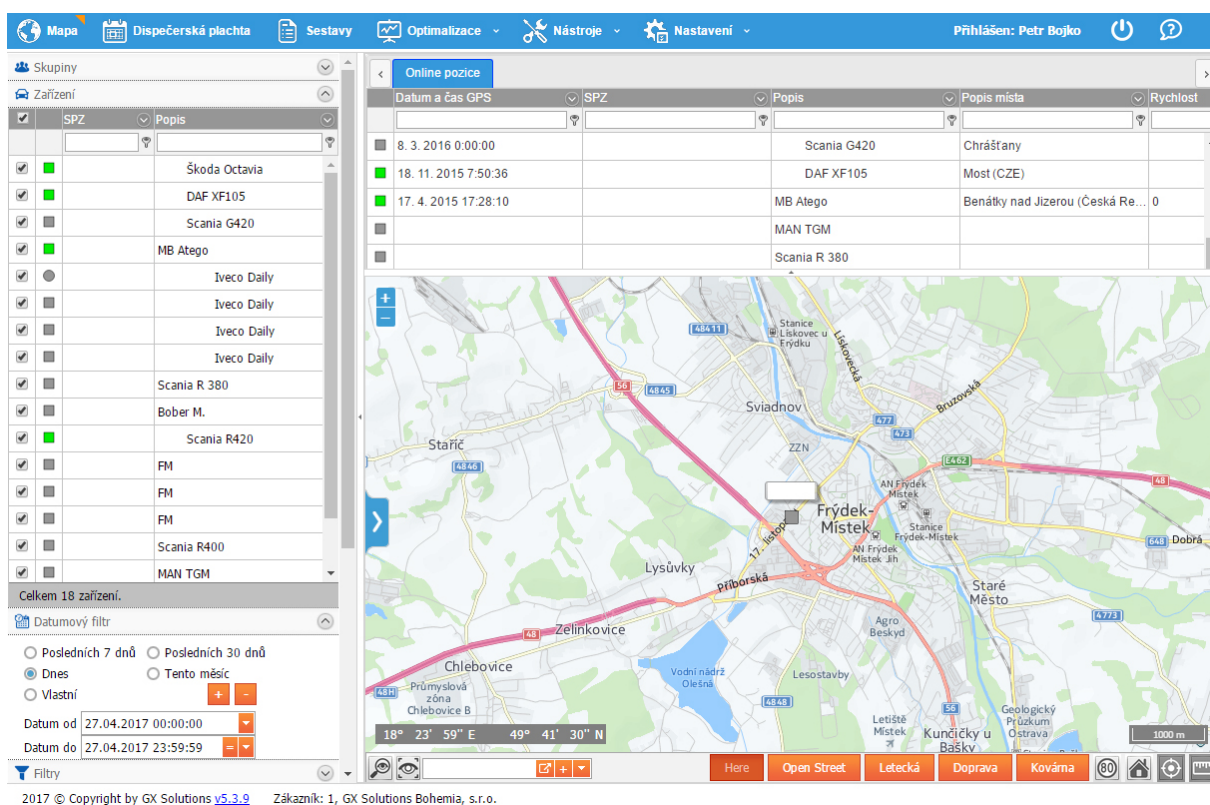
Mezi hlavní moduly systému patří mapový modul, který poskytuje informace především o poloze vozidel a jejich stavu, mimojině také funkce jako vyhledání trasy za použití *IT4Innovations* routingu.

Dispečerskou plachtu, kterou uživatelé mohou využít pro zápis plánovaných událostí do časové osy.

Modul sestav, který slouží ke generování souhrnných informací za určitý časový interval a popřípadě export a tisk dokumentů založených na získaných datech.

Dalším z modulů je modul hlídání událostí, pomocí něhož si může uživatel nadefinovat, při kterých stavech vozidel má dostat upozornění. Jako příklad je možné uvést překročení nadefinované rychlosti v uživatelem určené oblasti na mapě.

Systém samozřejmě obsahuje i již zmíněný modul optimalizace, který bude více popsán v samostatné kapitole 7.



Obrázek 2: Mapový modul GX-GO

4 Použité technologie

4.1 WCF

Windows Communication Foundation je jednotný programovací model od společnosti *Microsoft*. Jedná se o technologii vyvinutou za účelem budování aplikací orientovaných na služby. Vývojáři díky ní mohou vyvíjet bezpečné, spolehlivé a multiplatformní řešení. [4]

4.2 ASP.NET

Tento webový framework je součástí technologie *.NET*. Tato sada knihoven umožňuje vývojáři vytvářet webové aplikace v jazyce *C#* a *Visual Basic .NET*. Knihovny obsahují již hotová řešení pro zabezpečení a šifrování, autentifikaci uživatele, práci s databází, správu formulářů a mnoho dalších. *ASP.NET* nabízí dva způsoby jak vytvářet webové aplikace. Je možné využít *ASP.NET WebForms*, kdy se webová aplikace tváří jako desktop aplikace, nebo *ASP.NET MVC*, pomocí které je vyvinuta i aplikace, o které pojednává tato práce. [5]

4.3 ASP.NET MVC

Tato technologie pracuje s konceptem, který rozděluje webovou aplikaci do tří komponent, které jsou *Model*, *Controller* a *View*. *Controller* je řídicí komponenta, která se stará o data přijatá od uživatele a předá je komponentě *Model*, která obsahuje business logiku a komunikuje s databází. Data, která *Controller* získal od *Model* pak přepoše komponentě *View*, která je zobrazí na výsledné *HTML* stránce. [5]

4.4 IIS Express

IIS Express je služba, která umožňuje vytvořit webový server, pomocí kterého lze provozovat a testovat webové stránky. Podporuje protokoly jako jsou HTTP, HTTPS, FTP, FTPS, SMTP a NNTP. [7]

4.5 Razor syntaxe

Razor syntaxe usnadňuje tvorbu *HTML* šablon za pomoci jazyka *C#* nebo *Visual Basic .NET*. Podobně jako u *ASPX* je možné zapisovat kusy kódu přímo do *HTML* stránky. [21] Funkční kód se uvozuje znakem `<%` a porovnání lze vidět na následujících úryvcích kódu. [22]

```
<ul id="products">
    <% foreach (var p in products) { %>
        <li><%=p.Name%> ($<%=p.Price%>)</li>
    <% } %>
</ul>
```

Výpis 1: ASPX syntaxe

```
<ul id="products">
  @foreach (var p in products) {
    <li>@p.Name ($@p.Price)</li>
  }
</ul>
```

Výpis 2: Razor syntaxe pro C#

4.6 DevExpress pro ASP.NET MVC

Jedná se o sadu tříd a nástrojů od společnosti Developer Express Inc, která slouží k vývoji webových aplikací. Tato sada poskytuje rozšířenou funkcionalitu ASP.NET a nástroje pro vizuální práci s komponentami. [13]

4.7 PostgreSQL

Multiplatformní systém pro správu objektově-relačních databází s otevřeným zdrojovým kódem. Splňuje všechny podmínky ACID, podporuje uložené procedury, trigger a spoustu dalších funkcí. Zahrnuje také většinu datových typů dle standardu SQL:2008. Podporuje také ukládání obrázků, zvuků i videa. [10]

Databáze je využita především pro ukládání optimalizovaných tras společně s informacemi týkajícími se např. přiřazeného řidiče a posloupnost souřadnic vygenerované trasy uloženými v poli bajtů.

Limit	Hodnota limitu
Maximální velikost databáze	Neomezená
Maximální velikost tabulky	32 TB
Maximální velikost řádku	1,6 TB
Maximální velikost pole	1 GB
Maximální počet řádků tabulky	Neomezená
Maximální počet sloupců tabulky	250 - 1600*
Maximální počet indexů tabulky	Neomezená

* záleží na typech sloupců

Tabulka 1: Limity PostgreSQL

4.8 OpenLayers

OpenLayers je open source technologie fungující na Javascriptu, uvolněná pod licencí *FreeBSD*. Slouží k možnosti zobrazení dynamických map na webové stránky. Umožňuje zobrazit mapové

dlaždice, vektorová data a nebo značky získané z jakéhokoliv zdroje. Je využita k potřebám této práce, konkrétně k vykreslení trasy navracené z VRPTUO do mapy. [20]

4.9 JavaScript a jQuery

JavaScript je interpretovaný, objektově-orientovaný jazyk využívaný především pro vytváření skriptů webových stránek. Pro jednodušší vytváření webových stránek byla vyvinuta JavaScript knihovna jQuery, která ulehčuje vytváření animací, ošetřování událostí, AJAX a spoustu dalších. [8] [9]

4.10 Entity Framework

Jde o sadu technologií, které podporují vývoj aplikací orientovaných na data. Jedná se o objektově-relační mapper. Entity Framework umožňuje vývojářům pracovat s daty ve formě doménově specifických objektů a vlastností. Díky němu se tedy nemusí potýkat přímo s tabulkami, sloupci a jejich daty. Pomocí Entity Framework jsou vývojáři schopni pracovat na vyšší úrovni abstrakce, než když se zabývají daty a mohou vytvářet aplikace pomocí méně řádků kódu, než u aplikací vyvinutých pomocí tradičních metod. [6]

5 Příklad obchodního cestujícího

V rámci zadání diplomové práce byl jedním z bodů nastudování Problému obchodního cestujícího.

Tento problém, zvaný také TSP (Travelling Salesman Problem) je jednou z úloh kombinatorické optimalizace, jež má za cíl nalézt v ohodnoceném grafu kružnici takovou, která prochází všemi vrcholy a její cena je minimální. Jinými slovy se tato optimalizace snaží nalézt nejkratší hamiltonovskou kružnici v ohodnoceném grafu.

Jedná se o problém NP-úplný a silně NP-obtížný.

V praxi může sloužit k řešení problémů jako nalezení nejlevnější cesty, pokud jsou uzly grafu města, hrany představují cesty mezi městy ohodnocené na základě jejich vzdálenosti a je za potřebí nalézt nejlevnější cestu mezi všemi městy a zpět do výchozího města, pokud cena cesty přímou úměrou závisí na vzdálenosti cesty.

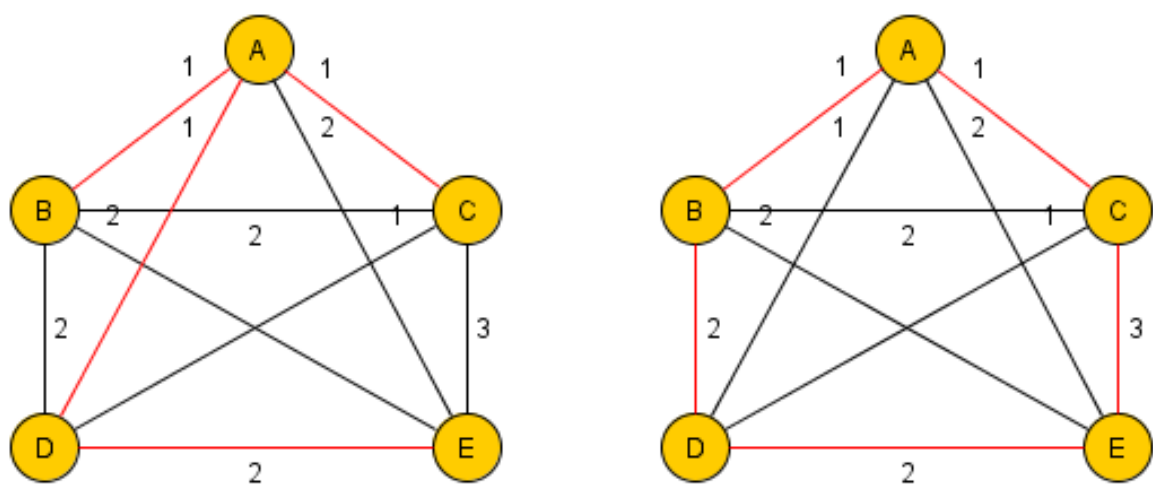
5.1 Metrický obchodní cestující

Tento problém je variantou TSP, kdy se předpokládá, že ohodnocené hrany grafu splňují trojúhelníkovou nerovnost, tj. součet délek dvou libovolných stran je vždy větší, než délka strany třetí. Díky tomuto předpokladu je umožněna konstrukce aproximačních algoritmů.

5.1.1 2-aproximační algoritmus

Tento algoritmus vede k řešení Metrického obchodního cestujícího. Dosahuje toho tím, že nejdříve konstruuje minimální kostru grafu, dále projde kostru z libovolného uzlu do hloubky a poznamená si všechny průchody přes vrcholy a jako poslední krok provede zkrácení cest tak, že vynechá všechny duplicity (zanechání pouze prvních výskytů uzlů). [18]

Jak lze vidět na obrázku 3, nejdříve je provedeno nalezení minimální kostry grafu, poté prohledání grafu do hloubky a nakonec odstranění duplicitních cest.



Obrázek 3: Příklad 2-aproximačního algoritmu
[19]

6 Vehicle routing problem

Tento algoritmus řešící problém optimalizace trasy pro potřeby přepravních společností byl v roce 1959 navržen Georgem Ramtzigem a Johnem Ramserem. [23]

Cílem je, aby dopravní flotila společnosti byla schopna obhospodařit co nejvíce požadavků a zároveň splnila všechny podmínky, které plynou z kapacitních omezení vozidla, případně řidiče apod.

Jedná se o zobecnění Problému obchodního cestujícího. Stejně jako TSP může být VRP reprezentován jako ohodnocený graf, kde ceny hran představují vzdálenosti mezi městy. VRP může narozdíl od TSP navíc obsahovat určitá omezení jako:

- Váha, objem, počet palet pro danou objednávku,
- maximální počet vyřízených objednávek pro dané vozidlo,
- časová okna, mezi kterými musí být objednávka vyřízena,
- časová okna, která představují řidičovu pracovní dobu a další.

[24]

Tento problém je tedy mnohem sofistikovanější než TSP, jelikož se musí vzít v úvahu spousta dalších ovlivňujících faktorů.

Existuje několik různých variant, jako následující:

- CVRP - obsluha jedním vozidlem s omezenou kapacitou,
- SDVRP - obsluha s více vozidly,
- VRPTW - omezení časových oken,
- MDVRP - vozidla mohou pocházet z různých dep 11.1,
- VRPPD - vozidlo může kdykoliv vykládat i nakládat zboží,
- a mnohé další varianty.

[25]

Pro potřeby společnosti je nutné počítat s několika variantami řešení tohoto problému, z důvodu mnoha parametrů, které může zákazník v rámci užívání systému nastavit. Ať už je to obsluha s více vozidly, tak fakt, že vozidla mohou být z různých dep a také jsou limitována časovými okny a samozřejmě nosností vozidel a objemem daných objednávek.

Řešení tohoto problému je realizováno v současné době za pomoci modifikované verze jsprit knihovny, ale společnost *GX Solutions* z důvodu spolupráce s *IT4Innovations* rozhodla o přechodu na jejich řešení VRP, a to knihovnu VRPTUO.

Tento krok nevyžadoval extrémní zásahy do chování systému, bylo ale nutné provést několik změn, co se týče přeformátování získaných dat do patřičné podoby, se kterou počítají dané části systému.

7 Současný stav optimalizačního modulu

Druhý bod zadání této práce se týkal nastudování současného stavu optimalizačního modulu v systému GX-GO za účelem lepšího pochopení problematiky a případně rozpoznání limitu a nebo navržení nutných změn.

Jak již bylo zmíněno výše, v současném stavu modul optimalizace používal jsprit knihovnu převedenou do jazyka C#.

Modul optimalizace obsahuje v rámci svého rozhraní pět stěžejních částí, a to následující:

- tabulku vozidel, které se mohou podílet na optimalizaci trasy objednávek,
- mapový sektor, který slouží pro vykreslení vygenerované trasy, ať už vzdušných tras nebo za použití dat routingu,
- tabulku již optimalizovaných tras, obsahující informace o počtu zastávek, době trvání jízdy, nákladech apod.,
- tabulku objednávek, které budou v rámci trasy vyřízeny,
- tabulku všech vytvořených objednávek.

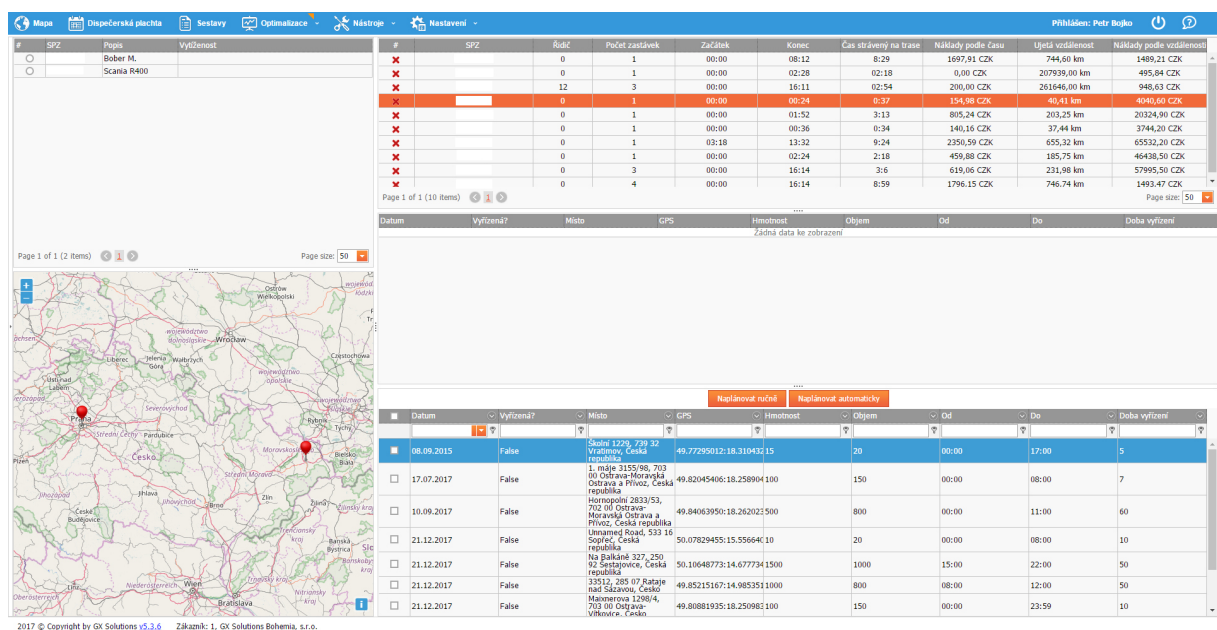
Viz obrázek 4.

Uživatel má možnost skrze nastavení aplikace zvolit následující parametry týkající se vozidel a řidičů:

- Pozice a názvy svých dep, 11.1
- příslušnost vozidla k danému depu,
- přiřazení řidiči vozidla,
- časové okno, kdy může řidič řídit,
- limity vozidla jako nosnost a objem nákladního prostoru,
- cena za hodinu práce řidiče, popřípadě za ujetý kilometr vozidla.

Dále, každá objednávka, kterou zákazník skrze modul optimalizace vytvoří, může, popřípadě musí obsahovat:

- Váhu,
- objem,
- časové okno od a do,
- doba vyřízení objednávky (čas vykládky, popřípadě vyřízení fakturace apod.).



Obrázek 4: Modul optimalizace

8 Modul AETR v GX-GO

V systému GX-GO je tzv. modul AETR, jelikož se ale společnost zaměřuje na zákazníky ze zemí Evropské Unie, modul dodržuje konkrétně vyhlášku ES 561, podle způsobu, který ve své bakalářské práci popsal autor modulu AETR Martin Kössler. [15]

8.1 Tachograf

Je to zařízení sloužící ke kontrole doby řízení vozidel tak, aby nebyly porušovány vyhláška AETR, popřípadě ES 561. Přítomnost tachografu je vyžadována u všech vozidel nad 3.5 tuny. V přístroji je zaznamenávána mimo jiné rychlost vozidla v závislosti na době jízdy.

Existují dvě varianty tachografů, a to analogové a digitální. V případě analogových tachografů jsou informace o řízení ukládány na papírové záznamové kotouče a u digitálních na vestavěnou paměť tachografu a na čipovou kartu řidiče.

8.2 Vyhlášky AETR a ES 561

Jak již bylo zmíněno v úvodu, vyčerpání řidičů hraje velkou roli při vzniku dopravních nehod. Tomuto problému se snaží zamezovat tyto dvě vyhlášky.

Samotná vyhláška AETR umožňuje, aby si každá země svá pravidla modifikovala a tím pádem by řidič, který by přešel hranice České republiky musel dodržovat pravidla zahraniční země. I když se tato práce řídí vyhláškou ES 561 musel být tento fakt brán v úvahu.

Z důvodu toho, že *IT4Innovations* routing poskytuje informace o trasách jen v rámci České republiky, byl tento fakt možných odlišností v pravidlech ostatních zemí zamítnut, jelikož by algoritmus nebyl schopen určit, za jak dlouho by byl řidič trasy v sousedních zemích odjet.

V Evropské unii a několika dalších zemích se od roku 2007 jezdí podle nařízení ES 561, který je téměř identický jako AETR.

Řidičův stav je tedy sledován pomocí tachografu. V rámci něj mohou být řidiči ve 4 různých stavech a to:

- Řízení
- Práce
- Pohotovost (přestávka)
- Odpočinek

Stavy jako řízení a pohotovost mohou být tachografem zaznamenávány automaticky tak, že při nastartování vozidla dojde k přepnutí do stavu řízení a při zastavení přejde k přechodu do stavu pohotovosti. Další stavy ale záleží na řidičově manuálním přepnutí.

Pokud se jedná o plně neautomatizovaný tachograf, je celá režie na řidiče a data jsou plně závislá na jeho poctivosti.

Základní pravidla a pojmy, které jsou v rámci modulu zohledňovány jsou následující:

- *Doba řízení* se chápe jako celková doba řízení od okamžiku, kdy řidič začne řídit vozidlo do doby, kdy začne další přestávka nebo odpočinek.
- *Denní doba řízení* je celková doba řízení mezi skončením jedné denní doby odpočinku a začátkem následující denní doby odpočinku, popřípadě začátkem týdenní doby odpočinku.
- *Týdenní doba řízení* je celková doba řízení ohraničená začátkem a koncem týdne.
- *Přestávka v řízení* je doba, během níž řidič nevykonává žádnou práci, ať už ve formě řízení nebo jakékoliv jiné.
- *Doba odpočinku* je nepřerušovaná doba, během které má řidič právo volně nakládat se svým časem dle svého uvážení.
- *Denní doba odpočinku* je doba odpočinku v rámci daného dne, která může přesahovat až do dne následujícího. Dělí se na *běžnou denní dobu odpočinku*, což je odpočinek trvající minimálně 11 hodin, který lze rozdělit do dvou časových úseků z nichž první musí být trvající minimálně 3 hodiny a druhý minimálně nepřerušovaných 9 hodin.
- *Zkrácená denní doba odpočinku* je doba odpočinku, která trvá méně než 11 hodin, ale více než 9 hodin.
- *Týdenní doba odpočinku* je doba, kdy řidič volně nakládá se svým časem a dělí se na *běžnou týdenní dobu odpočinku* a *zkrácenou týdenní dobu odpočinku*.
- *Běžná týdenní doba odpočinku* musí být v délce trvání nejméně 45 hodin.
- *Zkrácená týdenní doba odpočinku* je doba kratší než 45 hodin odpočinku, která může být zkrácená na nejméně 24 hodin, pokud bude do konce třetího následujícího týdne dostatečně vykompenzována.

Další z časových omezení popisujících přímo konkrétních časy, které modul AETR zohledňuje:

- Denní doba řízení nesmí být delší než 9 hodin, s tou výjimkou, že dvakrát během jednoho týdne smí být prodloužena až na 10 hodin.
- Týdenní doba řízení nesmí být delší než 56 hodin.
- Celková doba řízení za období dvou po sobě jdoucích týdnů nesmí přesáhnout 90 hodin.
- V případě, že řidiči zrovna nezačíná některá z dob odpočinku, ať už denní nebo týdenní, je nutné, aby po 4,5 hodinách jízdy měl 45 minut přestávku, jež může být rozdělena do dvou a to v délce nejméně 15 a 30 minut.

- Mezi dvěma týdenními dobami odpočinku smí mít řidič maximálně tři zkrácené denní doby odpočinku.

Data, která modul AETR je schopen uživateli pro daného řidiče navrátit a která jsou z části využívána i pro optimalizaci trasy jsou viditelná na obrázku 5.

	Denní statistiky	Týdenní statistiky
Doba řízení:	30:10(max. 9:00)	60:52(max. 44:35)
Čas do odpočinku:	0:00	0:00
Min. délka příštího odpočinku:	9:00	24:00
Počet prodloužených jízd:	2/2	
Počet zkrácených odpočinků:	0/3	
Časová známka:	27.04.2017 18:57:51	
Zpráva:	Příliš dlouhá denní doba řízení, Příliš dlouhá týdenní doba řízení	

Obrázek 5: Data získaná v rámci modulu AETR

9 Routing v GX-GO

Routing pro GX-GO zajišťuje služba od *IT4Innovations*. Ve webové aplikaci je routing používán pro generování trasy pro potřeby uživatelů, kdy je zadán startovní a koncový bod a uživateli je navracena výsledná trasa s posloupností všech názvů ulic, kterými v rámci trasy potenciální vozidlo projede.

Toto je spojeno s načtením více než 200 MB souboru, který tyto informace obsahuje, do paměti. Z tohoto důvodu bylo první vyhledávání trasy delší o dobu načítání souboru.

Po zvážení bylo rozhodnuto, že tyto informace pro potřeby optimalizace trasy nejsou absolutně nutné a tím pádem je možné čekací dobu na načtení souboru eliminovat díky vytvoření alternativní verze routingové metody, která potřebné informace získá.

Metoda na navrácení dat routingu nám umožňuje specifikovat jak zeměpisnou šířku a délku počátečního bodu, tak zeměpisnou šířku a délku cílového bodu. Mimo to, umožňuje zadat i parametry, které určí, zda vyhledaná trasy má umožnit průjezd skrze placené úseky, skrze úseky určené i pro kamiónovou dopravu a zda má být během hledání trasy upřednostněna její rychlost nebo délka.

Dále je možné předat i informace týkající se maximální rychlosti, váhy a výšky vozidla.

```
Result route = service.GetRoute((float)poi.Latitude, (float)poi.Longitude, (
    float)neighbor.Latitude, (float)neighbor.Longitude, false, true, true, 0,
    0, 0);
```

Výpis 3: Kód zisku dat routingu (C#)

Jak lze vidět na obrázku 6, výsledek routingu obsahuje i data konkrétních názvech ulic a délku na nich strávenou v rámci dané trasy. Jak bylo zmíněno výše, tato data nejsou pro optimalizaci trasy podstatná a lze proto ušetřit nemalé množství času při prvním načtení.

The screenshot shows a web interface for route calculation. At the top, there are two input fields: 'Z' (Start) with the value '17. listopadu 912, Místek, 738 01 Frý' and 'Do' (End) with the value 'Petrov nad Desnou 229, 788 14 Petro'. Below these is a link '+ Přidat průjezdný bod'. There are two vehicle icons: a red car and a black truck. Below the icons are two radio buttons: 'Nejkratší' (unselected) and 'Nejrychlejší' (selected). There is a checked checkbox 'Placené úseky'. At the bottom are two orange buttons: 'Vyhledat trasu' and 'Vymazat'. Below the buttons, the route summary is displayed: '177 km - 2:41 h'. At the very bottom, there is a table with route segments:

Začátek: 17. LISTOPADU	409 m
R56	9.6 km
MÍSTECKÁ	12.5 km

Obrázek 6: Modul routingu

V rámci tohoto modulu nebylo třeba žádné další analýzy, jelikož všechny potřebné informace byly zpracovány.

10 Popis postupu optimalizace a návrh začlenění celkového řešení

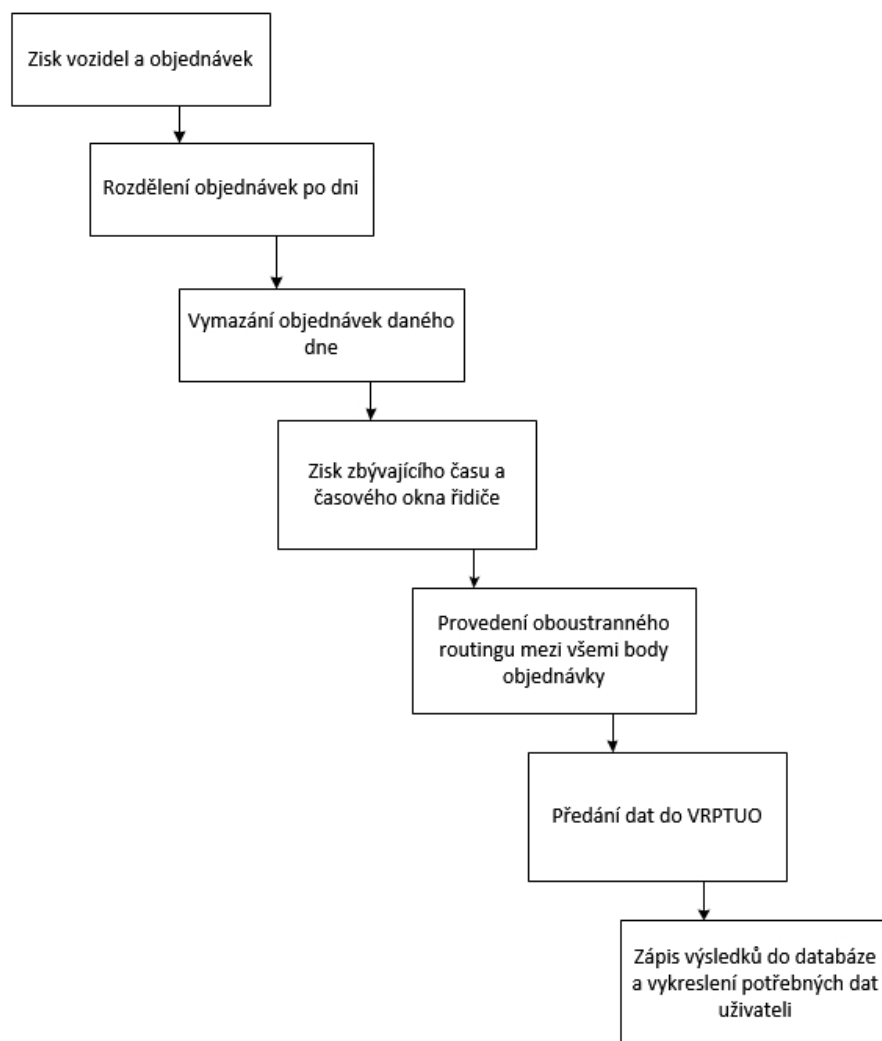
Před samotným návrhem řešení routingu a dat AETRu bylo nutné zvolit vhodné umístění provedení těchto kroků. Uživatel po výběru objednávek má na výběr ze dvou možností a to, zda použít automatický výběr nejvhodnějšího vozidla, nebo výběr provede sám a pro jedno dané vozidlo bude podstatným výstupem z optimalizace jen pořadí navštívených bodů mapy a zda je vůbec schopné všechny objednávky v rámci svých kapacit zvládnout.

Ať už uživatel vybral jakoukoliv alternativu, dalším krokem, který se provede je rozdělení vybraných objednávek podle dne, pro který byly zadány. Pokud bude zjištěno, že v daný den již existují nějaké zadané objednávky, budou anulovány.

V tuhle chvíli již přichází na řadu zisk řidiče nebo řidičů vozidla. Pro každého z nich se získá jeho zbývajících čas a časové okno během kterých může obsloužit dané objednávky, více kapitola 12. Na základě toho, který řidič má nejvyšší jízdní čas se určí řidič daného vozidla pro optimalizaci.

Po zisku nejvhodnějšího řidiče každého vozidla se přechází do fáze, kdy jsou pro všechny body objednávek daného dne a pro všechna depa daných vozidel získány vzdálenostní trasy více v kapitole 11.

Po předání dat do VRPTUO se provede optimalizace a navracená data jsou uložena v databázi a navracena uživateli, viz obrázek 7



Obrázek 7: Návrh postupu řešení v základních bodech.

11 Návrh začlenění routingu do optimalizace

V tomto kroku byl brán ohled na co nejširší pokrytí možných problémů spojených s ovlivněním odježdění trasy.

V rámci analýzy možnosti konfigurace vozidel, které uživatel při používání *GX-GO* má, byla zjištěna možnost specifikování maximální rychlosti každého vozidla ze zákaznickovy flotily.

Jelikož je schopnost odjet dané trasy přímo závislá na trvání jízdy a na zbývajícím čase v rámci vyhlášky ES 561, zvažovala se varianta, kdy se zákaznickova vozidla rozdělí do skupin na základě nastavení svých maximálních rychlostí, a poté bude provedeno načteční dat routingu pro každou skupinu zvlášť a tyto informace předány do VRPTUO.

Z důvodu vysokých čekacích dob na výsledek routingu byla tato možnost zamítnuta, a proto jsou data a vozidla omezena vždy maximální rychlostí komunikace, přes kterou právě projíždí.

11.1 Depo

Výchozí bod pro vozidlo společnosti. Na tomto místě vozidlo naloží dané zboží a po dokončení trasy se, pokud si uživatel nepřeje jinak, vrací zase zpět na toto místo.

11.2 Vyloučení použití vzdálenostní matice

Z důvodu větší rychlosti byla GX Solutions nabídnuta vzdálenostní matice mezi městy České republiky od *IT4Innovations*.

Po zvážení bylo rozhodnuto, že tato data nejsou dostatečná, jelikož neposkytují přesnou informaci o tom, jak dlouho trvá trasa mezi dvěma body mapy.

Určení časového trvání trasy na základě konstantní rychlosti a délky trasy nemůže být uspokojivý, a to z důvodu měnících se rychlostí úseků trasy mezi dvěma body na mapě.

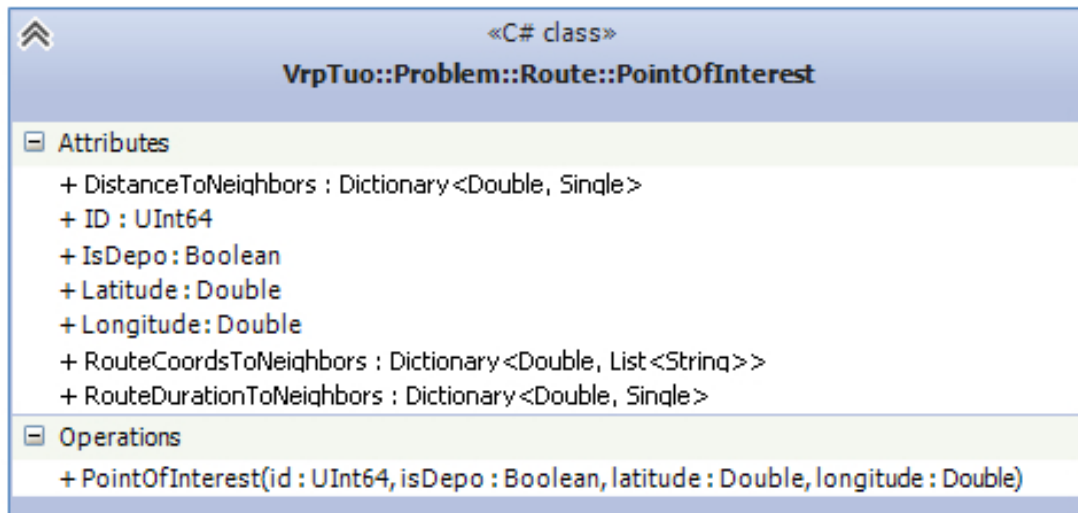
Vozidla jedné společnosti mohou být rozdělena mezi více dep.

11.3 Model pro data routingu

Model obsahuje i informace, které v současném stavu nejsou využívány, např. informace, zda se jedná o depo, nebo jen bod objednávky viz obrázek 8. VRPTUO bude pracovat především se dvěma slovníky a to *DistanceToNeighbors*, který obsahuje klíč dané lokace a k němu z daného bodu a *RouteDurationToNeighbors*, který obsahuje klíč a hodnotu v podobě času stráveného na trase.

11.4 Paralelní zisk dat routingu

Z důvodu nedostatečné rychlosti služby routingu a velkého množství cest, které je při větším množství objednávek a dep nutno získat bylo využito paralelního přístupu získání potřebných dat. Jelikož VRP problém potřebuje znát vzdálenosti mezi všemi body a jejich sousedy je tedy nutné aby mezi každými dvěma body bylo provedeno získání trasy. Toto samotné ale nestačí.



Obrázek 8: Třídní diagram bodu zájmu (depo/místo objednávky)

Na rozdíl od normálního ohodnoceného grafu, kde je mezi dvěma uzly, představujícími místo na mapě jen jedna cesta s jednou cenou, v reálném světě tento přístup možný není.

Cesta z bodu A do bodu B a zpět z bodu B do bodu A se u větších vzdáleností může lišit i o několik stovek metrů z důvodů odlišnosti různých směrů komunikací.

Proto bylo nutné provést požadavek na data routingu oboustraně pro každý bod mapy a jeho sousedy. Paralelně pro každý bod na mapě (depo nebo objednávku) se provede paralelní vyhledávání tras ke všem jeho sousedům a potřebné informace se uloží do datového modelu.

```

var service = new RoutingService();
var neighbors = pois.Where(w => w.ID != poi.ID).ToList();
var distanceToNeighbors = new ConcurrentDictionary<double, float>();
var routeDurationToNeighbors = new ConcurrentDictionary<double, float>();
var routeCoords = new ConcurrentDictionary<double, List<string>>();
Parallel.ForEach(neighbors, (neighbor) =>
{
    Result route = service.GetRoute((float)poi.Latitude, (float)poi.Longitude, (
        float)neighbor.Latitude, (float)neighbor.Longitude, false, true, true, 0,
        0, 0);
    distanceToNeighbors.TryAdd(neighbor.Latitude, route.LengthInMetres);
    routeDurationToNeighbors.TryAdd(neighbor.Latitude, route.TimeInSec);
    routeCoords.TryAdd(neighbor.Latitude,
        route.Segments.Select(s => string.Format(System.Globalization.CultureInfo.
            CreateSpecificCulture("en-US"), "{0}:{1},", s.LatitudeA, s.LongitudeA)).
            ToList());
});

```

```

poi.DistanceToNeighbors = distanceToNeighbors.ToDictionary(kvp => kvp.Key, kvp
=> kvp.Value);
poi.RouteDurationToNeighbors = routeDurationToNeighbors.ToDictionary(kvp => kvp
.Key, kvp => kvp.Value);
poi.RouteCoordsToNeighbors = routeCoords.ToDictionary(kvp => kvp.Key, kvp =>
kvp.Value);

```

Výpis 4: Paralelní získání dat routingu (C#)

11.5 Předání dat do VRPTUO

Po získání všech potřebných dat routingu jsou předána do VRPTUO. Knihovna si v případě výpočtu vzdušné vzdálenosti nikde neuchovávala již vypočítané vzdálenosti z důvodu velice rychlých výpočtů.

Tento přístup v případě routingu není v současné době realizovatelný z důvodu rychlosti a ani nevhodný, z důvodu zbytečně vysokého počtu dotazů na servery *IT4Innovations*.

Proto bylo nutné vytvoření statického slovníku ze kterého si VRPTUO vybralo patřičné hodnoty na základě toho mezi jakými dvěma lokacemi zrovna počítala vzdálenost. Pokud slovník není naplněn znamená to, že zákazník si přeje použít vzdušné vzdálenosti a VRPTUO provede výpočet bez dat routingu. Viz výpis kódu 4.

```

const double D = 100000d;
Dictionary<double, float> poiDistances = pois.Where(w => Convert.ToInt32(w.
Value.Latitude * D) == precept.A.X && Convert.ToInt32(w.Value.Longitude * D
) == precept.A.Y).Select(s => s.Value.DistanceToNeighbors).FirstOrDefault()
;

Dictionary<double, float> poiDurations = pois.Where(w => Convert.
ToInt32(w.Value.Latitude * D) == precept.A.X && Convert.ToInt32(
w.Value.Longitude * D) == precept.A.Y).Select(s => s.Value.
RouteDurationToNeighbors).FirstOrDefault();
if (poiDistances != null)
{
    routingDistanceInMeters = (double)poiDistances.Where(w => Convert.ToInt32(w.
Key * D) == precept.B.X).Select(s => s.Value).FirstOrDefault();
    journeyDurationHours = (double)TimeSpan.FromSeconds(poiDurations.
Where(w => Convert.ToInt32(w.Key * D) == precept.B.X).Select(s =>
s.Value).FirstOrDefault()).TotalHours;
}
else

```

```
{  
    routingDistanceInMeters = distanceInMeters;  
    journeyDurationHours = (routingDistanceInMeters * 0.001) / precept.Vehicle.  
        MaxVelocity;  
}
```

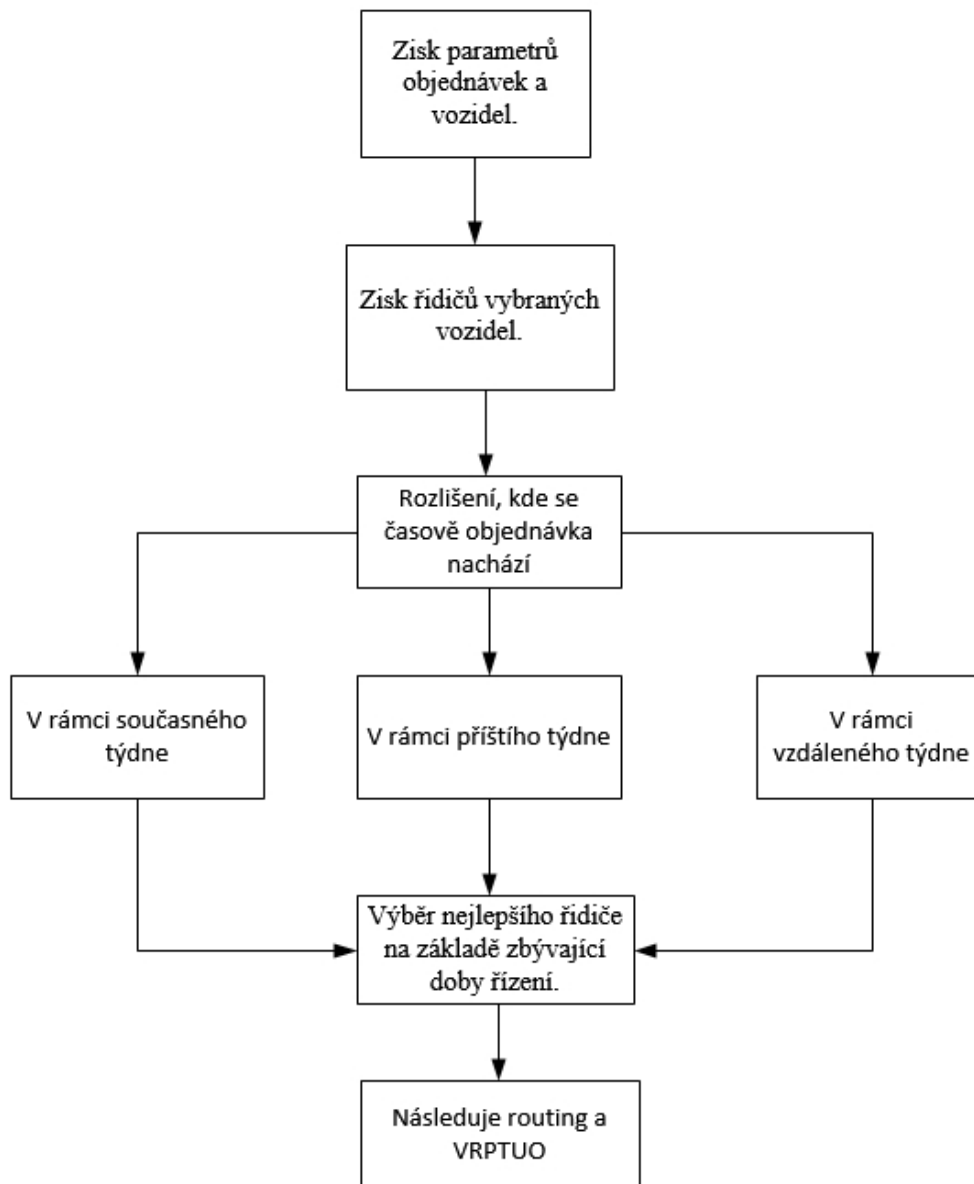
Výpis 5: Zisk patřičných vzdáleností mezi dvěma body (C#)

12 Návrh začlenění AETRu do optimalizace

Tento bod je stěžejní pro celou práci. Od začátku byl brán ohled na to, aby modul kontroly dat na základě AETRu byl co nejvíce samostatný a nezávislý na běhu samotného VRPTUO. Toto ale vede k nedostatkům, které byly objeveny až ve fázi, kdy přepracování nebylo možné stihnout v souladu s termínem.

Bylo navrženo řešení, které je v další kapitole 14 blíže popsáno.

Základní postup operací, které vedou k zohlednění ES 561 je načrtnut na obrázku číslo 9.



Obrázek 9: Návrh začlenění zohlednění ES 561

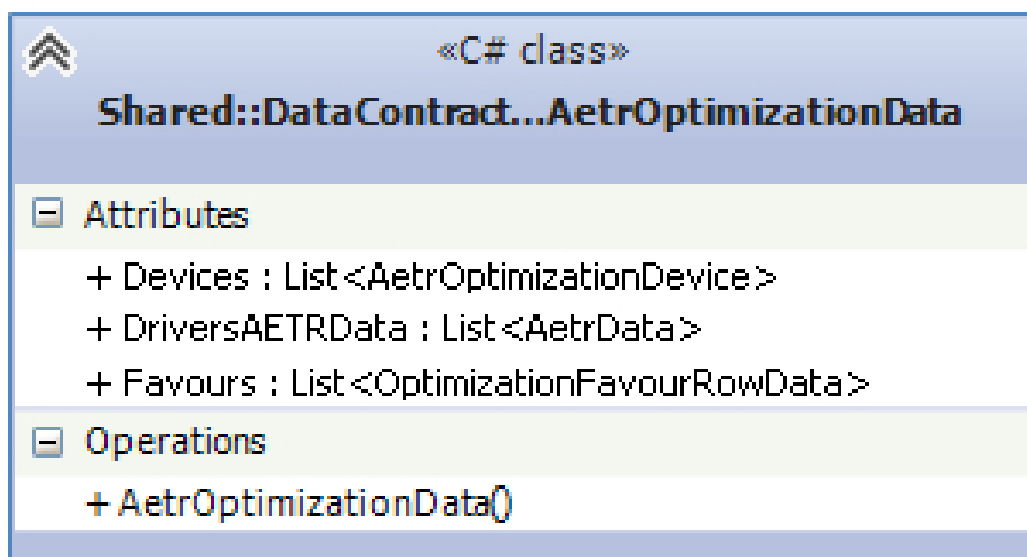
12.1 Vyčítání řidičů

Jak již bylo zmíněno, zákazník má možnost nadefinovat si řidiče, popřípadě více řidičů ke svým vozidlům. Tato informace jde tedy zjistit buď ze záznamu v databázi, nebo případně se dá vyčíst na základě posledního přihlášeného řidiče z jednotky vozidla.

Pokud nebyl nalezen pro dané vozidlo ani jeden řidič, vozidlo se vyřadí, jelikož bez řidiče není možné zjistit stav zbývajcího jízdního času. Pokud bylo nalezeno více řidičů

12.2 Datový model pro výsledek zohlednění AETRu

V tomto případě bylo nutné získat pro daného řidiče dvě hlavní informace a to jak údaj a době jízdy, kterou může pro daný den odjezdit, tak údaj, kdy bude pro danou jízdu připraven, např. kdy bude ukončen denní odpočinek z minulého dne viz obrázek 10.



Obrázek 10: Varianta dopočítání zbývajcího jízdního času pro současný týden

12.3 Dopčet budoucích dat o zbývajcí jízdní době

Pokud by si uživatel přál zoptimalizovat jízdy pro, které nezačínají v daný den, nemůže být vyhodnocení AETRu provedeno jen na základě dat získaných z modulu AETR.

Proto bylo nutné zohlednit tři možné případy, které mohou nastat.

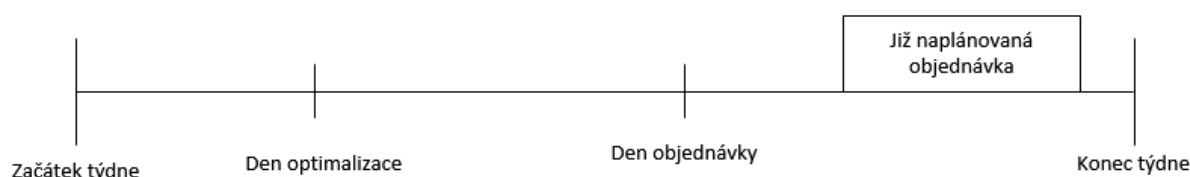
Kvůli požadavkům společnosti a zákaznické logiky plánování objednávek, kdy se ví o všech denních objednávkách dopředu nebylo nutné zohledňovat objednávky v den optimalizovaných tras. To znamená, že pokud uživatel má již naplánované trasy na nějaký den, neočekává se, že by k nim přidával nějaké další. Pokud ano, již naplánované trasy se smažou a tím pádem

není nutné, odečítat dobu jízdy strávenou jejich prováděním pro denní dobu jízdy pro novou optimalizovanou objednávku.

12.3.1 Jízdní čas v rámci současného týdne

Pokud se datum optimalizovaných objednávek nachází v rámci současného týdne, máme možnost pracovat s reálnými daty ES 561, které nám již existující modul je schopen navrátit. Toto ale není dostačující.

Jak již bylo zmíněno v kapitole Modul AETR v GX-GO 8, data o současném stavu jsou získávány na základě již provedených událostí, modul ale nepočítá s již naplánovanými jízdami daného řidiče, které mohou v daném týdnu nastat. Pro lepší představu dané situace viz 11.



Obrázek 11: Varianta dopočítání zbývajcího jízdního času pro současný týden

Jak lze z obrázku rozpoznat, data z modulu AETR, která máme v den optimalizace nepočítají s tím, že do konce týdne je pro daného řidiče již naplánovaná závazná objednávka, kterou musí odřídit a mohly by způsobit vyčerpání týdenního jízdní doby a v případě zahrnutí dalších objednávek řidiči by už mohlo dojít k porušení vyhlášky.

Proto je nutné v rámci algoritmu vzít i tyto již naplánované objednávky a jejich čas vzít v úvahu.

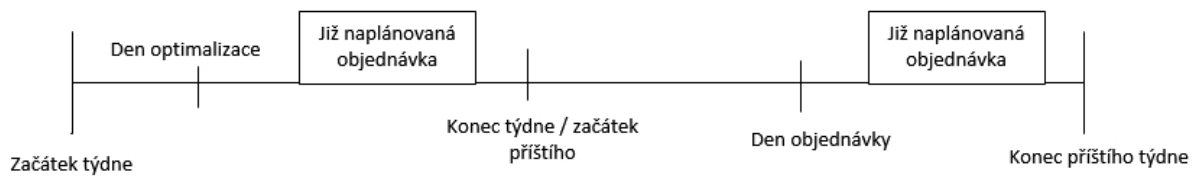
12.3.2 Jízdní čas v rámci následujícího týdne

Tato možnost přichází v úvahu, že plánované objednávky jsou v příštím týdnu od týdne, kdy probíhá optimalizace.

V tomto případě je možnost získat data o změnách stavů v rámci ES 561 jen v současném týdnu, k nim dále přidat informace o jízdách, které v rámci již naplánovaných tras daný řidič provede, a poté vyhledat jízdy naplánované v rámci příštího týdne.

Poté jsou tato data předána úpravené verzi metody modulu pro zpracování dat ES 561 pro získání zbývajcího jízdního času s tím rozdílem, že jako výchozí okamžik není použitý současný den, ale den optimalizovaných objednávek.

Celá tato situace je načrtnuta na obrázku 12, kde lze vidět jak současný týden, kde v den objednávky máme reálná data zaznamenaná tachografem, tak objednávky současného i příštího týdne, které musíme všechny do výpočtu zahrnout.



Obrázek 12: Varianta dopočítání zbývajících jízdního času pro příští týden

12.3.3 Jízdní čas v rámci vzdálené budoucnosti

Tato varianta již nemůže spoléhat na data získaná v rámci změn stavů pomocí tachografů. V tomto případě je nutné spoléhat čistě jen na délky již zadaných objednávek pro daného řidiče. Proto je nutné získat všechny dané jízdy v časovém okruhu týdne, kdy je daná objednávka zadána a také týdne před ní samotnou. V kódu dané metody je připravená i varianta, kdy se již naplánované jízdy daného dne nebudou anulovat a budete se počítat s jejich jízdními časy pro nově přidávanou objednávku daného dne, viz 15.



Obrázek 13: Varianta dopočítání zbývajících jízdního času pro příští týden

```

if (thisWeek.Count == 0 && prevWeek.Count == 0)
    return Utils.GetSecondsFromHours(10);

long prevWeekDriveSeconds = 0;
long twoWeeksMaxDriveSeconds = Utils.GetSecondsFromHours(90);

if (prevWeek.Count != 0)
    prevWeekDriveSeconds = prevWeek.Sum(s => s.Duration);

twoWeeksMaxDriveSeconds -= prevWeekDriveSeconds;

if (twoWeeksMaxDriveSeconds > Utils.GetSecondsFromHours(56))
    twoWeeksMaxDriveSeconds = Utils.GetSecondsFromHours(56);

```

```

Dictionary<DateTime, List<PersonHistoryAetr>> daysDrivesGroups = thisWeek.
    GroupBy(g => g.Since.Date).ToDictionary(kvp => kvp.Key, kvp => kvp.ToList()
    );

int extendedDayDriveCount = 0;
long thisWeekDriveSeconds = 0;
for (int i = 0; i < daysDrivesGroups.Count(); i++)
{
    long dayDrivingSeconds = daysDrivesGroups.Values.ElementAt(i).Sum(s => s.
        Duration);
    thisWeekDriveSeconds += dayDrivingSeconds;
    if (dayDrivingSeconds > (9 * 3600))
        extendedDayDriveCount++;
}

if (thisWeekDriveSeconds >= Utils.GetSecondsFromHours(twoWeeksMaxDriveSeconds))
    return 0;

long thisDayMaxDriveSeconds = Utils.GetSecondsFromHours(10);

if (extendedDayDriveCount > 2)
    thisDayMaxDriveSeconds = Utils.GetSecondsFromHours(9);

var thisDayDrives = daysDrivesGroups.ContainsKey(timeOfFirstFavour.Date) ?
    daysDrivesGroups[timeOfFirstFavour.Date] : null;

if (thisDayDrives != null)
    for (int i = 0; i < thisDayDrives.Count; i++)
        thisDayMaxDriveSeconds -= thisDayDrives[i].Duration;

return thisDayMaxDriveSeconds;

```

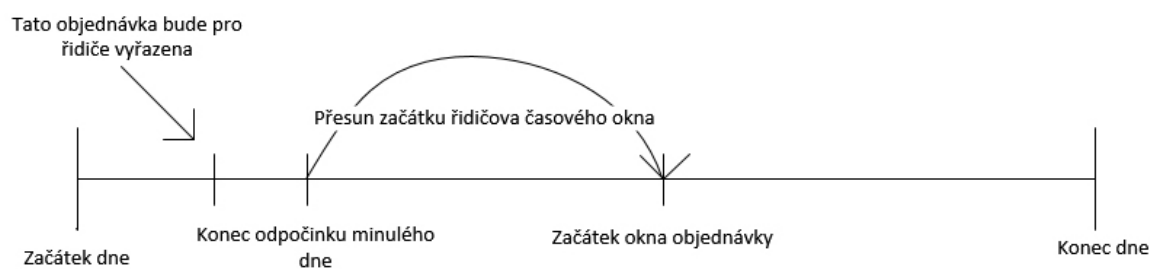
Výpis 6: Zisk budoucího jízdního času (C#)

12.4 Nastavení časového okna řidiče

Jak lze vidět na obrázku 14 konec denního odpočinku z minulého dne skončil až po začátku objednávky, jejíž časové okno začíná v daném dni nejdříve. Proto je vysoká pravděpodobnost, že by vozidlo nebylo vyhodnoceno jako nejvhodnější pro odjetí dané objednávky.

Z toho důvodu a také proto, aby nedošlo čekáním na začátek časového okna následující objednávky k jeho vyčerpání, které je založené na denní době jízdy, je začátek řidičova okna přesunut až na čas začátku časového okna objednávky.

Tento přístup by bylo vhodné optimalizovat způsobem, který by umožnil nastavit řidiči jeho časové okno začínající dříve o dobu nutnou k dojetí k dané objednávce tak, aby byl již v okamžik začátku časového okna objednávky na místě.



Obrázek 14: Posun časového okna k časově nejbližší následující objednávce

12.5 Úprava nastavení optimalizace

V rámci těchto nově zavedených funkcí je uživateli umožněno si přes nastavení aplikace vybrat, zda požaduje optimalizaci trasy za pomoci dat z routingu, nebo za pomoci leteckých vzdáleností a zda si při optimalizaci přeje zohlednit stav zbývajících jízdního času svých řidičů, viz obrázek 15.

The screenshot shows a form titled 'Nastavení výpočtu trasy'. It contains two radio buttons: 'Použít routing' (selected) and 'Použít leteckou vzdálenost'. Below these are three checkboxes: 'Použít placené úseky?' (unchecked), 'Zohlednit jízdní dobu řidičů?' (unchecked), and 'Zohlednit styl jízdy řidiče:' (unchecked). At the bottom is an orange button labeled 'Uložit'.

Obrázek 15: Nastavení modulu optimalizace

13 Testování na datech zákazníka GX Solutions

K testování funkčnosti úprav provedených v modulu optimalizace GX-GO bylo použito reálných dat o vozidlech zákazníků a popř. dat získaných modulem AETR.

Jelikož ale modul optimalizace není zákazníkům zatím uveřejněn, nebylo možné vycházet z reálných objednávek a bylo proto nutné objednávky pro optimalizaci vytvořit.

13.1 Rozdíly mezi optimalizací s a bez dat AETRu

Pro nasimulování této situace byly vytvořeny tři objednávky ve dvou po sobě jdoucích dnech.

První objednávka byla nastavena na 20.12.2017 s časovým oknem mezi 18:00 a 23:00. Byla přiřazena vozidlu s řidičem číslo 12. Ten se vrátil zpátky na depo ve 22:09 viz 16, v ideálním případě by tedy měl mít 11 hodinový odpočinek trvající do 09:09.

Objednávky následujícího dne mají časová okna od 00:00 do 08:00 a od 15:00 do 22:00. Jak lze vidět na obrázku 16 na třetím řádku tabulky, algoritmus rozpoznal, že začátek časově nejbližší následující objednávky po řidičově odpočinku je až v 15:00, proto mu začátek jeho časového okna posunul na tento čas. První objednávka dne 21.12.2017 měla konec svého časového okna v 08:00, z tohoto důvodu ji řidič neměl šanci realizovat a byla přiřazena řidiči číslo 2. V případě, že by ale objednávka měla začátek časového okna v 08:00 a konec např. ve 12:00, řidič by ji pravděpodobně také zvládl obsloužit, ale z důvodu, že by toto mohl udělat až po více než hodinovém zpoždění oproti řidiči číslo 2, byla jeho prioritní objednávkou ta pozdější.

Řidič	Počet zastávek	Začátek	Konec
12	1	18:00	22:09
2	1	00:00	05:53
12	1	15:00	16:31

Obrázek 16: Optimalizované objednávky za použití AETRu

V případě že zohlednění AETRu bude vypnuto, výsledky budou velice odlišné. Jelikož není řidič číslo 12 omezen denním odpočinkem z minulého dne, byl schopen vyřídit obě objednávky daného dne a VRPTUO knihovna ho vyhodnotila jako nejvhodnějšího kandidáta. Viz obrázek 17

Řidič	Počet zastávek	Začátek	Konec
12	1	18:00	22:09
12	2	00:00	16:15

Obrázek 17: Optimalizované objednávky bez použití AETRu

13.2 Rozdíly mezi optimalizací s a bez dat routingu

V rámci testování byla porovnávána optimalizovaná trasa za použití dat získaných díky routingu *IT4Innovations*, tak za použití výchozího chování VRPTUO, a to počítání na základě vzdušných vzdáleností.

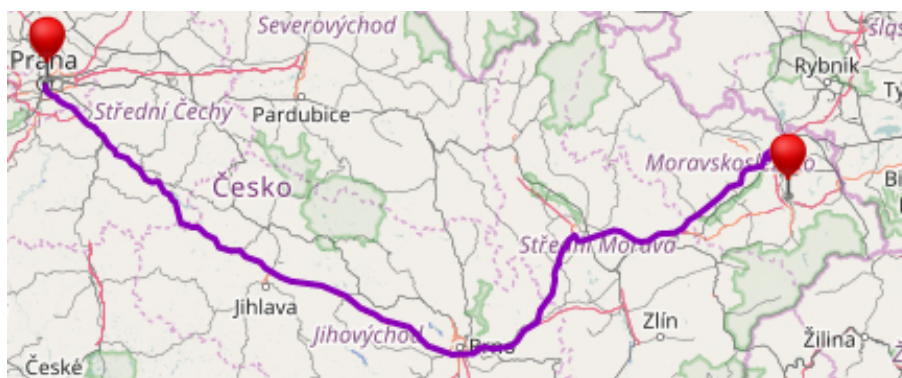
Na obrazcích číslo 18 a číslo 20 lze vidět, jak odlišnosti v délce i trvání trasy, tak přirozeně i ve vygenerované trase na mapovém podkladu 19 21. Na pořadí objednávek neměla tato změna v tomto případě vliv z důvodu minimálního dopadu rozdílu ve vzdálenostech, a také velice odlišných časových oknech mezi objednávkami, které neumožňovaly, aby se projevilo jiné pořadí navštívených bodů na mapě. Rozdíly lze vidět také v nákladech podle ujeté vzdálenosti.

Řidič	Počet zastávek	Začátek	Konec	Čas strávený na trase	Náklady podle času	Ujetá vzdálenost	Náklady podle vzdálenosti
12	1	18:00	22:09	04:04	200,00 CZK	366,22 km	833,93 CZK
2	1	00:00	05:53	05:43	25000000,00 CZK	549,65 km	202152800,00 CZK
12	1	15:00	16:31	00:41	200,00 CZK	57,44 km	306,08 CZK
12	1	00:00	08:19	07:19	200,00 CZK	753,48 km	1665,63 CZK
0	1	00:00	00:24	0:37	154,98 CZK	40,41 km	4040,60 CZK
0	1	00:00	01:52	3:13	805,24 CZK	203,25 km	20324,90 CZK
0	1	00:00	00:36	0:34	140,16 CZK	37,44 km	3744,20 CZK

Page size: 50

Místo	GPS	Hmotnost	Objem	Od	Do	Doba vyřízení
-------	-----	----------	-------	----	----	---------------

Obrázek 18: Optimalizace trasy za pomoci dat z routingu



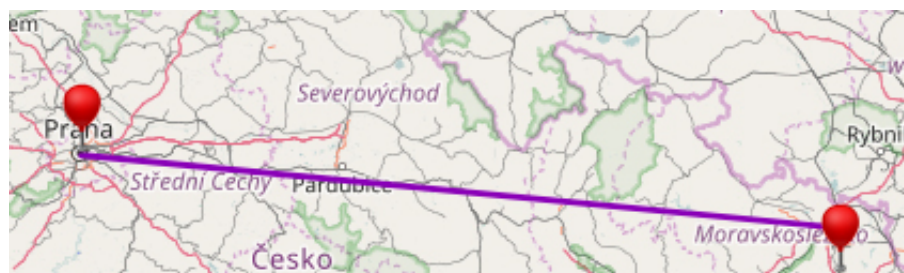
Obrázek 19: Trasa na mapě za pomoci dat z routingu

13.3 Testování s reálnými objednávkami zákazníků

V současné době probíhá testování na reálných datech objednávek zákazníků společnosti zadávaných v rámci jednoho z modulů desktop aplikace. Výsledky těchto testů zatím ale nejsou dostupné.

Řidič	Počet zastávek	Začátek	Konec	Čas strávený na trase	Náklady podle času	Ujetá vzdálenost	Náklady podle vzdálenosti
12	1	18:00	22:09	04:04	200,00 CZK	366,22 km	833,93 CZK
2	1	00:00	05:53	05:43	25000000,00 CZK	549,65 km	202152800,00 CZK
12	1	15:00	16:31	00:41	200,00 CZK	57,44 km	306,08 CZK
12	1	00:00	09:31	08:31	200,00 CZK	767,82 km	1907,80 CZK
0	1	00:00	00:24	0:37	154,98 CZK	40,41 km	4040,60 CZK
0	1	00:00	01:52	3:13	805,24 CZK	203,25 km	20324,90 CZK
0	1	00:00	00:36	0:34	140,16 CZK	37,44 km	3744,20 CZK

Obrázek 20: Optimalizace trasy za pomoci dat vzdušných vzdáleností



Obrázek 21: Trasa na mapě za pomoci dat vzdušných vzdáleností

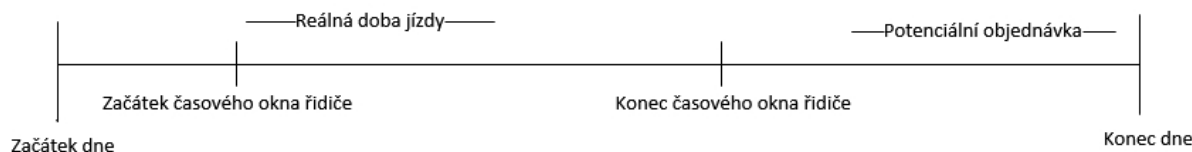
14 Omezení algoritmu

Jelikož bylo řešení zohledňování ES 561 v rámci této práce mířeno na řešení v podobě samostného modulu došlo k různým omezením.

Největším vzniklým problémem, který se projevil je omezení času jízdy řidiče za pomoci časového okna, které určuje, kdy může řidič obsloužit objednávky a zvládnout se vrátit na depo. Tento čas je přímo závislý na zbývajícím jízdě době, ale nepočítá s tím kolik v rámci trasy bude reálně věnován čas řízení.

14.1 Velice rozlišná okna objednávek

Tento problém se může projevit především v případě rozlišných časových oken objednávek. Pokud se časové okno první objednávky nachází mezi časem od 00:00 do 10:00 a druhé okno od 20:00 do 22:00, může nastat případ, kdy řidič první objednávku vyřídí v 05:00 a je schopen do 10:00 dojet k druhé objednávce, kde by měl možnost do 20:00 provést zkrácenou denní dobu odpočinku. Současné řešení tuto situaci není schopno rozpoznat a mohla by nastat situace lépe viditelná na obrázku 22.



Obrázek 22: Nedostatek v rámci nastavení časového okna

Jedním z možných řešení by bylo post testování navráceného výsledku z VRPTUO.

V tomto případě by se testovalo, jaká byla reálná doba jízdy řidiče a poté by byla porovnána se zbývajícím jízdě časem daného řidiče. Na základě této informace by se provedlo upravení časového okna řidiče a nové spuštění VRPTUO optimalizace, které by po dokončení muselo znovu projít testem, zda nebyla žádná z jízdě dob řidičů překročena.

Mnohem vhodnějším řešením by bylo integrování nového parametru vozidla/řidiče v knihovně VRPTUO. Tento parametr by fungoval na podobném principu jako například nosnost vozidla. Problém nastává ve chvíli, kdy se třída rozhoduje, zda danou objednávku vozidlu přiřadit nebo ne.

V případě nosnosti vozidla se jedná o relativně primitivní problém, kdy stačí pouze váhu další objednávky přiřadit k již naloženému množství vozidla a testovat, zda nebyla nosnost překročena.

U zbývajícím času je toto jiné. To, zda vozidlo je schopno k dané objednávce a zpátky do depa dojet v rámci zbývajícím jízdě času záleží na kterém konkrétním bodě v rámci mapy se nachází. Pokud by bylo stále v depu, může cesta trvat 2 hodiny, pokud by se ale nacházelo na místě některé z jiných objednávek, cesta k oné objednávce by mohla trvat dobu úplně jinou.

V tomto případě by se musela sledovat aktuální pozice vozidla a na základě toho, by se od jeho zbývajících času odečetla doba trasy ze současného bodu, k případné další objednávce.

Protože jsou tyto případy, kdy by bylo okno jedné objednávky například od 00:00 do 10:00 a druhé od 20:00 do 22:00, v praxi velice nepravděpodobné, toto omezení by v reálném použití nemělo nastat.

14.2 Možné nepovolené navýšení jízdního času řidiče

Jelikož je k jízdní době každého řidiče před předáním k optimalizaci do VRPTUO navýšena jízdní doba v podobě časového okna o čas potřebný k vykládce každé objednávky, může se stát, že řidiči bude umožněno odřídit více objednávek, než mu umožňuje vyhláška.

Pro lepší pochopení je možné si představit, že zákazník zadá tři objednávky, kdy dvě z nich mají nastavenou dobu vykládky 50 minut a třetí má dobu vykládky 10 minut. Díky tomuto získá řidič ke svému časovému oknu, odvozenému od jeho zbývajících jízdní doby celých 110 minut navíc, které by v rámci VRPTUO mohly znamenat, že řidiči bude přiřazena třetí objednávka, kterou by byl schopen s tímto časem navíc odjet i navzdory vlivu 10 minut na vykládku.

14.3 Problém s různými rychlostmi vozidel

Jak bylo řečeno již v kapitole Návrh začlenění routingu do optimalizace 11, vozidla mohou mít jiné maximální rychlosti, definované přímo zákazníkem. Jelikož data z routingu a délka jízdy po dané trase jsou přímo spojené s maximální rychlostí dané komunikace, může nastat situace, kdy bude očekáváno, že vozidlo danou trasu ujede za čas dostačující, aby nedošlo k porušení podmínek a maximální jízdní době, ale v realitě odjede trasu mnohem později z důvodu omezení své rychlosti.

V praxi je velice nepravděpodobné, aby tyto rozdíly, způsobené omezením rychlosti vozidla, byly markantní. Přesto by bylo vhodné v dalších krocích rozšíření této práce se na tuto problematiku zaměřit. Toto by mělo být umožněno díky nové verzi *IT4Innovations* routingu, která by měla poskytovat výsledky tras v nesrovnatelně kratším čase.

15 Možná rozšíření do budoucna

Kromě bodů zmíněných v kapitole Omezení algoritmu 14 by se v budoucnu mohl algoritmus rozšířit o následující funkce. Tyto funkce byly probrány s vedoucím této diplomové práce a jejich myšlenka i návrh řešení byly odsouhlaseny.

15.1 Zohlednění stylu jízdy řidiče

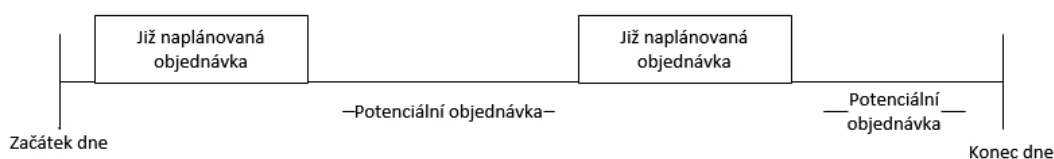
Jak lze vidět na obrázku 15, který znázorňuje možnosti nastavení průběhu optimalizace, mezi možnostmi nastavení je již zavedena možnost zohlednění stylu jízdy řidiče. Podle návrhu, který byl společnosti *GX Solutions* předložen by se v případě stejných, nebo velice podobných zbývajících jízdních časů řidičů, přihlédlo ke známce stylu jízdy.

Tato známka by byla vygenerována na základě dat za posledních 30 dnů od data optimalizace. V rámci této známky by se hodnotilo, zda řidič prudce brzdí, prudce se rozjíždí, nebo se například nepohybuje v optimálních otáčkách v rámci rychlosti vozidla. Pokud bude zákazníkovi v budoucnu umožněno, aby k vytvoření každé objednávky přidal i informaci o tom, zda se jedná o křehké zboží, mohl by tento rozhodovací faktor mít náležitě vysokou prioritu.

15.2 Překrývající se nové objednávky se starými

Logika užívání modulu optimalizace je v současné době taková, že zákazníci vždy plánují dané objednávky dopředu, a pokud se rozhodnou, že k danému dni je nutné ještě nějakou další objednávku přidat, všechny již naplánované trasy v rámci daného dne jsou anulovány.

Tento přístup ulehčil řešení zohlednění povolené doby jízdy o počítání s daty, která má již daný zákazník v daný den naplánované. V budoucnu by ale stálo za zvážení navržení algoritmu, který by již naplánované trasy nemazal, ale pokoušel by řidičovo časové okno nastavit co nejvyhodněji tak, aby mohl vyřídit nově zadané objednávky, viz obrázek 23.



Obrázek 23: Zvolení vhodného časového okna

15.3 Zkrácení přestávky

Řidič má v rámci pravidel ES 561 možnost mezi dvěma týdenními odpočinky zkrátit trvání své denní přestávky z 11 hodin na nejméně 9 hodin.

Tento fakt by algoritmus mohl vzít v úvahu, pokud by to bylo přínosné pro optimalizaci objednávek. V případě, že by jeho řádný denní odpočinek končil až 08:00 hodin dne objednávky

a daná objednávka by měla začátek svého okna v 07:00, mohlo by se přistoupit ke zkrácení denního odpočinku až o 2 hodin, pokud by na to měl řidič v rámci posledních dvou týdnů nárok. Pro přibližnou ilustraci je tento postup načrtnut na obrázku 24.



Obrázek 24: Zkrácení doby řidičova denního odpočinku

16 Závěr

Cílem této práce bylo zapracování dat routingu a zohlednění dat modulu AETRu pro modul optimalizace. Zatímco zohlednění routingu bylo provedeno v relativně dostatečné formě, aby poskytovalo přesná data, zohlednění dat modulu AETR má své nedostatky a taky obrovské potenciální možnosti na rozšíření, jak je zmíněno v kapitolách 14 a 15.

V rámci této práce byly tyto nedostatky popsány a byl poskytnut i návrh případného řešení. Realizace těchto kroků bude provedena v následujících měsících.

Problematika AETRu je velice široký pojem, který zahrnuje spoustu pravidel, která se musí vzít v úvahu, z tohoto důvodu tato práce neposkytuje „konečné řešení“. Jejím cílem bylo aspoň částečně vyloučit možnost, že dopravce, popřípadě jeho dispečer za pomoci modulu optimalizace přiřadí množství objednávek řidiči, které by nemohl vyřídit, aniž by neporušil legislativu.

Literatura

- [1] *Telematika* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <https://www.fleetmatics.com/what-is-telematics/>
- [2] *O Firmě / GX Solutions* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <http://www.gxsolutions.cz/o-firme/>
- [3] *Fáze životního cyklu produktu* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <https://support.microsoft.com/cs-cz/lifecycle?C2=12905>
- [4] *Windows Communication Foundation* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <https://msdn.microsoft.com/en-us/library/dd456779%28v=vs.110%29.aspx>
- [5] *1. díl - Úvod do ASP.NET* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <http://www.itnetwork.cz/tutorial-uvod-do-asp-dot-net>
- [6] *Entity Framework* [online]. 2017 [cit. 2017-04-28]. Dostupné z: [https://msdn.microsoft.com/en-us/library/gg696172\(v=vs.103\).aspx](https://msdn.microsoft.com/en-us/library/gg696172(v=vs.103).aspx)
- [7] *IIS Express Overview : The Official Microsoft IIS Site* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <http://www.iis.net/learn/extensions/introduction-to-iis-express/iis-express-overview>
- [8] *JavaScript reference - JavaScript - MSDN* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <https://developer.mozilla.org/cs/docs/Web/JavaScript>
- [9] *jQuery* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <https://jquery.com/>
- [10] *PostgreSQL: About* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <http://www.postgresql.org/about/>
- [11] *ASP.NET Razor C# Syntax* [online]. 2017 [cit. 2017-04-28]. Dostupné z: http://www.w3schools.com/aspnet/razor_syntax.asp
- [12] *asp.net - Comparisons of Razor vs ASPX syntax - Stack Overflow* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <http://stackoverflow.com/questions/9699556/comparisons-of-razor-vs-aspx-syntax>
- [13] *Online Documentation - Developer Express Inc.* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <https://documentation.devexpress.com/#AspNet/CustomDocument7873>
- [14] *aetr-vs-561* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <http://www.centrum-preprav.cz/clanky-prilohy/aetr-vs-561.pdf>

- [15] KÖSSLER, Martin. *Absolvování individuální odborné praxe* [online]. Ostrava, 2016 [cit. 2017-04-28]. Dostupné z: <https://dspace.vsb.cz/handle/10084/116050>
- [16] FRIML, Jiří. *Modul optimalizace přeprav pro dopravní monitorovací systém* [online]. Ostrava, 2015 [cit. 2017-04-28]. Dostupné z: <https://dspace.vsb.cz/handle/10084/108805>
- [17] CHRISTOPH MANUEL MEYER. *Vehicle Routing under Consideration of Driving and Working Hours A Distributed Decision Making Perspective*. Wiesbaden: Betriebswirtschaftlicher Verlag Gabler, 2011. ISBN 9783834929426.
- [18] *Problém obchodního cestujícího* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <https://www.algoritmy.net/article/5407/Obchodni-cestujici>
- [19] 5449 [online]. 2017 [cit. 2017-04-28]. Dostupné z: <https://www.algoritmy.net/image/id/5449>
- [20] *Open Layers - Welcome* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <https://openlayers.org/>
- [21] *ASP.NET Razor C# Syntax* [online]. 2017 [cit. 2017-04-28]. Dostupné z: http://www.w3schools.com/aspnet/razor_syntax.asp
- [22] *asp.net - Comparisons of Razor vs ASPX syntax - Stack Overflow* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <http://stackoverflow.com/questions/9699556/comparisons-of-razor-vs-aspx-syntax>
- [23] *Vehicle Routing Problem / Vehicle Routing Problem* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <http://neo.lcc.uma.es/vrp/vehicle-routing-problem/>
- [24] *Vehicle Routing Problem / Optimization / Google Developers* [online]. 2017 [cit. 2017-04-28]. Dostupné z: https://developers.google.com/optimization/routing/tsp/vehicle_routing
- [25] *VRP* [online]. 2017 [cit. 2017-04-28]. Dostupné z: <https://www.fd.cvut.cz/projects/k611x1p/VRP.pps>

A Příloha na CD/DVD